

DataGRID at the ZMAW

User Manual

Max Planck Institute for Meteorology

April 24, 2006,
Docu Version: 0.2

Contents

1	What's a grid ?	1
2	The grid system at the ZMAW	1
3	Getting started	1
3.1	Before you can use the grid	1
3.2	Submit your first job	2
3.2.1	Use the command line	2
3.2.2	Use the graphical user interface QMON	3
4	Applications	3
4.1	Examples	3
4.1.1	gCCDAS : Second derivatives	4
4.1.2	gBETHY : Domain decomposition	4
4.1.3	gECHAM-post : Postprocessing of model output	4
.1	Gridenable your own application	5
.1.1	Wrapperscript	5
.1.2	Meta data	5
A	Submit hosts	6

1 What's a grid ?

There is a quite simple answer : A grid is a collection of computing resources, providing a transparent single point access.

There is a quite complex and sophisticated answer as well, see e.g. <http://en.wikipedia.org/wiki/> Furthermore there are different kinds and implementations of grid systems depending on what it should be good for.

2 The grid system at the ZMAW

The Sun Grid Engine (SGE) 'N1GE6' is installed and configured as a test configuration at the MPI for Meteorology and administrated by CIS. The SGE is a 'Distributed Resource Management' (DRM) tool, which accepts *jobs* - users' requests for computer resources. This jobs are put in a holding area (socalled *queue*) and send to appropriate worker nodes, as soon as they can be executed. They are managed during execution, and records are logged when they are finished. More detailed descriptions and manuals of the grid engine N1GE6 itself are provided on the Sun websites, e.g. <http://docs.sun.com/app/docs/coll/1017.3>. More informations about the cooperative project 'DataGrids in Earth System Sciences' can be found on <http://www.cis.zmaw.de/> in About Us → Projects → DataGrid

3 Getting started

After you are registered and have done a few settings, it is very easy to use the grid by submitting jobs.

3.1 Before you can use the grid

Some prerequisites must be fulfilled, before you can submit jobs to the grid.

1. **Become a grid user :** You must be registered as a *grid user* . In principle all users that have an account at the DKRZ can be registered. If you are not sure, whether you are already registered or you want to apply for registration, please fill out the form on the project website above or contact your grid administrator (gridadmin@dkrz.de).
2. **Log on a submit host :** After registration you must log in to a so called *submit host* , a computer, which allows for submitting and controlling batch jobs. You can find a list of all current submit hosts in the appendix. (If you want your own workstation to be a submit host, please mark the corresponding field in the registration form).
3. **Set the SGE environment variables:** The grid engine environment variables are set by the commands :
% `source /opt/gridware/sge/zmaw/common/settings.csh` for C-shell users
rsp.
% `. /opt/gridware/sge/zmaw/common/settings.sh` for Bourne shell users.
4. **Test your grid environment :** Check your settings by entering the command
% `qconf -ss`
All submit hosts should be listed.

Now you are ready for 'grid computing' !

3.2 Submit your first job

3.2.1 Use the command line

To test whether your grid environment is working, submit a simple test job from shell and check the execution status, e.g. :

```
% qsub /opt/gridware/sge/examples/jobs/simple.sh
```

Your job 721 ("simple.sh") has been submitted.

```
% qstat
```

```
job-ID prior name user state submit/start at queue
```

```
-----
```

```
721 0.55500 simple.sh m216015 r 07/26/2005 18:35:25
```

The job with job id 721 is running now (state 'r'). When the job is finished, it will no longer be listed by `qstat` and the output is written to a file.

There are several other job states, e.g.

- 'q': the job is queued, i.e. put in an appropriate job container (the queue)
- 'w': the job is waiting for execution
- 'E': an error occurred

Especially in the case of errors, you probably want to know, what happened to your submitted job. To get more detailed information enter

```
% qstat -f -j 721
```

 and check the messages there.

By default the output directory of the grid jobs is your home directory. Please check first the error output file `<jobname>.e<jobid>`, whereby jobname is in the above example `simple.sh` and the jobid is 721 :

```
% more $HOME/simple.sh.e721
```

If it exists and is empty (or just contains some timing or scheduling informations) all is fine. Then you should find the standard output of the respective job in

```
% more ~/simple.sh.o721
```

```
Wed Jul 27 11:38:26 CEST 2005
```

```
Wed Jul 27 11:38:46 CEST 2005
```

Congratulations, your grid job did as it should : write date and time, sleep for 20 seconds and again display date and time.

Otherwise, if

- your job stays in the state 'q' and/or 'w' for more than some minutes,
- it is in error state 'E',
- the error output `$HOME/<jobname>.e<jobid>` shows some error messages,
- or you get in any other trouble

and you can't get rid of these problems on your own, don't hesitate to contact the grid administrator at CIS.



Figure 1: QMON - Main control window

3.2.2 Use the graphical user interface QMON

Above we described, how you to submit a job by entering shell commands on a terminal. Some people prefer to work with a graphical user interface (GUI). The GUI for the Sun Grid Engine is called *qmon*.

Note : Unfortunately *qmon* does not work on all platforms at the moment. If you encounter problems on your machine, please contact the grid administrator.

If you call *qmon* according to `% qmon &`

the 'QMON +++ Main Control' window will pop up with a lot of buttons (see figure 1)

4 Applications

In this section we describe the implementation of more complex applications on the grid.

There are a few examples collected in the grid repository. The purpose of these examples is to demonstrate some features of the grid engine.

Once you feel confident concerning the functionality and use of the grid engine, you can 'wrap' your own applications in a job script and submit it to the grid.

4.1 Examples

In the repository `/opt/gridware/sge/applications` there are samples of grid applications as tar balls. To test and perform one of the applications you need to

1. log into a submit host,
2. copy the corresponding taarball to your working directory and unpack it,
3. carefully read the README file and follow the instructions

E.g. for the application *gCCDAS* :

```
% cp /opt/gridware/applications/gCCDAS.tgz .
% tar -zxpf gCCDAS.tgz
% cd gCCDAS
% more README
```

4.1.1 gCCDAS : Second derivatives

This example is taken from the project CCDAS (for details see <http://CCDAS.org>). Second derivatives of the cost function $J(x)$ w.r.t. the optimal model parameters x are used, to approximate the parameter uncertainties. The corresponding matrix

$$((H))_{i,j} = \frac{\partial^2 J}{\partial x_i \partial x_j} \quad i, j = 1, \dots, n \quad (1)$$

is called Hessian and has n^2 entries, where n denotes the number of entries.

As discussed in ? the computations of the individual columns of the Hessian are independent of each other, although there are saving when computing multiple columns in a simple job. But by parallelising the computations into several independant jobs the CPU time can be reduced significantly.

Once you have copied and unpacked the application package gCCDAS.tgz, please read the README file and follow the instructions.

4.1.2 gBETHY : Domain decomposition

The Biosphere Energy-Transfer and Hydrology model (BETHY) simulates exchange fluxes between the biosphere and the atmosphere for 'grid cells' of a 'global grid' ¹. For more details about the model see ?. The grid cells of this 'model grid' are treated entirely independent of each others, so we can separate the global grid into tails, i.e. groups of grid cells. Sometimes this is called 'domain decomposition'. The gBETHY application is an example of a very simple domain decomposition of BETHY, whereby the different tails are executed on different execution hosts.

Once you have copied and unpacked the application package gBETHY.tgz, please read the README file and follow the instructions.

4.1.3 gECHAM-post : Postprocessing of model output

We consider here a typical post processing workflow of ECHAM ² data at the ZMAW, which consists in four steps :

1. Get the 'raw data' from the output stream of a model run, an archive or out of a data base
2. The standard post processor for ECHAM output data is the "afterburner" providing several operations, as selecting codes, transform model grid (spectral to Gaussian) and write data in GRIB, NetCDF, SERVICE or EXTRA format.
3. Further postprocessing is done by 'CDOs' (Climate Data Operators) or the so called 'PINGOS'. E.g. extracting mean values from time series or selecting 'pressure level' from the atmospheric data.

¹Unfortunately the meaning of the term 'grid' is ambivalent here. In earth system models a grid denotes the segmentation of the globe surface in 'gridcells' whereas in IT technologies a grid is referred to as a 'service for sharing computer power and data storage'. To distinguish the different meanings we allways refer to former as 'model grid'

²ECHAM is an atmospheric general circulation model, for more details see <http://www.mpimet.mpg.de/en/extra/models/echam/>

4. Finally the data has to be stored in a data base or in an archive

The corresponding subtasks are assigned to the following four scripts, which are packed in the tar ball gECHAM-post.tgz :

1. gFTP.sh : This is not ECHAM specific, but executes just several ftp transfers on several execution nodes.
2. gAFTER.sh : The afterburner is called. In the current form just one code is selected from the given monthly RAW file.
3. gCDO.sh : The CDOs (or one of the PINGO programs) are called, to select specific codes and levels.
4. gFILL.sh : Should fill the data in the database. This is not implemented so far.

Remark : These post processing test scripts are by no way comprehensive and their only purpose is to give a base for your individual postprocessing and as a demonstration how the performance of typical postprocessing workflows can be improved.

.1 Gridenable your own application

Having shown how the grid can be used to improve workflows and processes of scientists you probably wonder, how you can make cabable your applications for the grid. The general answer is that you analyse your application for parts, which can be fractionalised in parts, which can be performed in parallel. Furthermore there are information needed, which specify the considered workflow with the used data, models and executables. These so called 'meta data' has to be provided by the user and can be used to wrap your application in a 'grid script'.

.1.1 Wrapperscript

We provide a template 'gTemplate' to simplify to wrap your application in a gridscrip.

.1.2 Meta data

Following meta data has to be provided by the users:

- application : name of the appl ...
- executable : this is the executable, that performs a sub task
- input data ...
- output ...

For the application examples above we get :

Application				
CCDAS				

A Submit hosts

We list here the current submit hosts at the ZMAW:

kurs01.zmaw.de
kurs02.zmaw.de
kurs04.zmaw.de
kurs05.zmaw.de
kurs06.zmaw.de
kurs07.zmaw.de
kurs08.zmaw.de
kurs09.zmaw.de
kurs10.zmaw.de
linde.mpi.zmaw.de
mccoy.cis.zmaw.de
pappel.mpi.zmaw.de
tanne.mpi.zmaw.de

Index

grid user, 1

Job, 1

qmon, 3

Queue, 1

submit host, 1